3주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

**1. Array 클래스 멤버 변수, 멤버 함수, 연산자 오버로딩**

**-멤버 변수**

int \* data : 배열 포인터

int len : data 배열 길이

**-멤버 함수**

1-1. Array(int size)

만약 size가 0보다 작으면 “Error: input negative” 에러 메세지를 출력하고,

아니라면 new 연산자로 size 크기의 int형 배열 data를 동적으로 선언하고,

len을 size로 업데이트한다.

1-2. ~Array()

소멸자는 delete 연산자를 통해 할당된 메모리를 해제하는데, 우리는 배열을 해제해야

하기 때문에 delete[] 를 사용한다. 마지막으로 len을 0으로 초기화한다.

1-3. int length() const

data 배열의 길이 len을 반환한다.

1-4. void print()

for문을 이용해 0~len까지 순회하면서 data배열의 각 원소를 출력한다.

**-오버로딩**

1-5. int& operator[](int i)

’[]’을 오버로딩하는 함수이며 data배열에 원소를 대입하는 함수이다.

static int tmp를 선언하고, 만약 i가 0과 같거나 크고 len보다 작을 경우 data[i]를 반환한다.

만약 아니라면 “Array bound error!” 에러 메세지를 출력하고, tmp를 반환한다.

1-6. int operator[](int i) const

’[]’을 오버로딩하는 함수이며 data배열의 원소를 참조하는 함수이다.

static int tmp를 선언하고, 만약 i가 0과 같거나 크고 len보다 작을 경우 data[i]를 반환한다.

만약 아니라면 “Array bound error!” 에러 메세지를 출력하고, 0을 반환한다.

**2. RangeArray 클래스 멤버 변수, 멤버 함수, 연산자 오버로딩**

**-멤버 변수**

int \* data : 배열 포인터

int len : data 배열 길이

**-멤버 함수**

2-1. RangeArray(int i, int j)

low에 i를 저장하고, high에 j를 저장한다.

그 다음 부모 클래스 Array 생성자 Array(int size)를 이용해서 Array(j-i+1)를 실행해

배열을 동적 할당한다. [ex: RangeArray(-1,5) 🡪 Array(5-(-1)+1) 🡪 size = 7]

2-2. ~RangeArray()

Array의 소멸자를 통해 할당된 메모리가 해제되기 때문에 함수 내부에 구현할 내용은 없다.

2-3. baseValue()

data배열의 low값을 반환한다.

2-4. endValue()

data배열의 high값을 반환한다.

**-오버로딩**

2-5. int& operator[](int i)

’[]’을 오버로딩하는 함수이며 data배열에 원소를 대입하는 함수이다.

Array 클래스 생성자를 통해 배열을 만들었기 때문에 배열의 범위는 0~(high-low)이다.

i에 음수도 포함되기 때문에 low값을 빼서 최솟값이 0이 되게 한다.

Array 클래스의 Overloading, Array::operator[](i-low)를 실행한다.

2-6. int operator[](int i) const

’[]’을 오버로딩하는 함수이며 data배열의 원소를 참조하는 함수이다.

2-5. 와 마찬가지로 Array 클래스의 Overloading, Array::operator[](i-low)를 실행한다.

**2.**

**1. Str 클래스 멤버 변수, 멤버 함수, 연산자 오버로딩**

**-멤버 변수**

char \* str : 문자열 포인터

int len : str 문자열 길이

**-멤버 함수**

1-1. Str(int leng)

만약 leng가 0보다 작으면 “Error: input negative” 에러 메세지를 출력하고,

아니라면 len을 leng으로 업데이트하고 new 연산자로 len 크기의 char형 배열 str을

동적으로 선언한다.

1-2. Str(char \*neyong)

strlen함수를 이용해 문자열 neyong의 길이를 len에 저장한다.

그 다음 len 크기의 char형 배열 str을 동적으로 할당한 뒤 strcpy함수를 이용해

neyong 문자열을 str에 복사한다.

1-3. ~str()

소멸자는 delete 연산자를 통해 할당된 메모리를 해제하는데, 우리는 배열을 해제해야

하기 때문에 delete[] 를 사용한다. 마지막으로 len을 0으로 초기화한다.

1-4. int length(void)

문자열의 길이 len을 반환한다.

1-5. int contents(void)

문자열의 내용 str을 반환한다.

1-6. int compare(class Str& a)

strcmp를 이용해서 문자열 str과 Str 클래스의 a객체의 문자열을 비교한 결과를 반환한다.

1-7. int compare(char\* a)

strcmp를 이용해서 문자열 str과 문자열 a를 비교한 결과를 반환한다.

**-오버로딩**

1-8. void operator=(char\* a)

’=’을 오버로딩하는 함수이며 문자열 a를 str에 대입하는 함수이다.

strcpy함수를 이용하여 str에 a를 복사하고,

len에는 strlen함수를 이용해서 문자열 a의 길이를 저장한다.

1-9. void operator=( class Str& a)

’=’을 오버로딩하는 함수이며 Str 클래스의 a객체의 문자열을 str에 대입하는 함수이다.

strcpy함수를 이용하여 str에 a.str을 복사하고,

len에는 strlen함수를 이용해서 문자열 a.str의 길이를 저장한다.